

Complétion automatique de mots (et gestion de dictionnaires)

Le but du projet est d'étudier et d'implémenter une ou plusieurs méthodes de complétion automatique de mots facilitant la saisie de texte.

La rédaction d'un document comporte très souvent la répétition des mêmes mots, l'ordinateur peut donc aider en proposant de compléter les mots, on parle d'écriture prédictive, comme sur les téléphones mobiles pour l'écriture de SMS (par exemple, le système T9). On se propose de réaliser un tel système pour créer ou compléter un fichier texte.

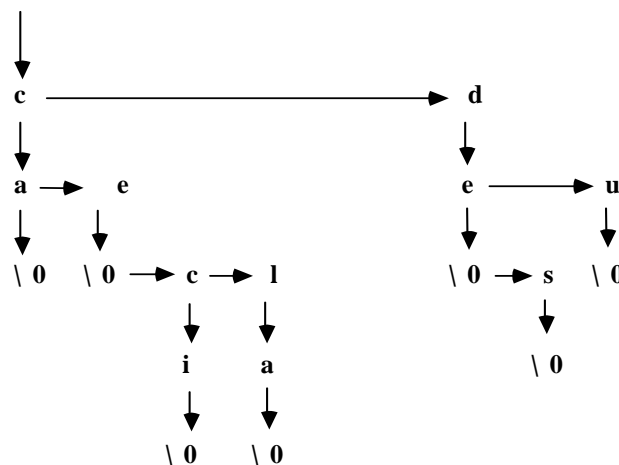
Le programme final doit permettre à l'utilisateur de taper du texte qui sera enregistré dans un fichier. Au fur et à mesure de la saisie d'une ligne, caractère par caractère, il détermine de façon intelligente l'ensemble ordonné des complétions possibles (les fins les plus probables du mot en cours de frappe), et en propose à l'utilisateur. Ce dernier choisit un des mots proposés, et le programme complète automatiquement le mot et lui permet de continuer la saisie de la ligne avec un nouveau début de mot. L'utilisateur peut aussi décider d'ignorer les propositions et de continuer la saisie des caractères du mot. Après la saisie d'un retour à la ligne ("entrée"), la ligne est écrite dans le fichier et le programme propose de taper une autre ligne. Etc.

Exemple :

```
Tapez votre ligne :  
un canard dit à sa ca  
vous pouvez compléter :  
0 cane  
1 canard  
2 canot  
3 canette
```

Une complétion est une terminaison possible du mot courant trouvée dans un dictionnaire préalablement chargé. Un dictionnaire sera un simple fichier texte. Pour des raisons d'efficacité (vous pourrez en discuter dans votre rapport), le dictionnaire sera représenté en mémoire sous forme d'un *arbre n-aire* (représentation "fils-frère"). Cette représentation ne stockera qu'une seule fois chaque préfixe de mot.

Exemple : le dictionnaire contenant les mots {ca, ce, ceci, cela, de, des, du} est représenté par l'arbre suivant.



Suivant le lien "frère", les lettres sont rangées par ordre alphabétique. De plus, pour indiquer les fins de mots, on utilise ici le caractère nul '\0'.

La solution proposée doit obligatoirement pouvoir gérer plusieurs dictionnaires et plusieurs méthodes de complétion.

Les méthodes de complétions à envisager sont :

- l'ordre alphabétique,
- des probabilités statiques,
- des fréquences d'utilisation dynamiquement calculées,
- ou toutes autres méthodes que vous définirez.

On pourra éventuellement proposer que les mots nouveaux (saisis par l'utilisateur et n'existant pas dans le dictionnaire chargé) soient automatiquement ajoutés à un dictionnaire secondaire "personnel" (lequel est chargé en complément du dictionnaire principal).

Travail à réaliser

- Etude détaillée des algorithmes liés à l'utilisation de l'arbre (insertion d'un mot, suppression, ...).
- Etude de différentes méthodes de complétion.
- Ecriture d'un programme dédié à la gestion de dictionnaires, permettant (par l'intermédiaire d'un menu texte) :
 - d'ouvrir et de fermer un dictionnaire,
 - d'afficher le dictionnaire chargé,
 - d'insérer un mot dans le dictionnaire chargé,
 - de supprimer un mot dans le dictionnaire chargé,
 - de rechercher un mot complet dans le dictionnaire chargé,
 - de rechercher un mot incomplet dans le dictionnaire chargé.
- Ecriture d'un deuxième programme de saisie de texte avec complétion de mots. Notez que, pour ce programme, les paramètres (noms de fichiers, ...) devront être fournis sur la ligne de commande (exemple : *progsaisietexte fichiersortie.txt ledico.txt ...*).

Les structures de données, les algorithmes de gestion du dictionnaire, et les méthodes de complétion sont les points importants de ce projet.